

In the Claims:

Please cancel claims 22-24, replace claim 1, and add new claim 25, all as shown below.

1. (Currently Amended): A dynamic code generation method for ~~dynamically generating program code~~, comprising:

~~dynamically generating program code, wherein dynamically generating program code includes:~~

creating a class file container object;

adding a method to the class file object;

adding code to the method using programming language constructs;

generating byte code for the class file container object; and

instantiating an instance of the new class file object;

generating executable code from the byte code by using a class loader; and

wherein dynamically generated code would be configured to exist for the life of a server it resides upon.

2. (Original): The method of claim 1 wherein creating a class file container object includes:
- setting attributes for a class file.

3. (Previously Presented): The method of claim 2 wherein the attributes include a class file name and a parent super class.

4. (Original): The method of claim 1 wherein adding a method to the class file object includes:

adding a plurality of methods to the class file object.

5. (Previously Presented): The method of claim 1 wherein the programming language constructs correspond to programming language statements, expressions, and variables.

6. (Previously Presented): The method of claim 5 wherein the programming language constructs include parameters.

7. (Previously Presented): The method of claim 5 wherein each statement, expression type, and variable is represented as an object.

8. (Previously Presented): The method of claim 1 wherein generating byte code for the class file container object includes:

generating an intermediate representation of program flow.

9. (Previously Presented): The method of claim 8 wherein generating byte code for the class file container object includes:

converting the intermediate representation into byte code.

10. (Original): The method of claim 1 wherein the ~~program~~ dynamically generated code implements an adaptor class.

11. (Original): The method of claim 1 wherein the program dynamically generated code implements a proxy class.

12. (Previously Presented): The method of claim 1 further comprising:
repeatedly adding a method to the class file object for each method associated with a stub generated for a remote object.

13. (Previously Presented): The method of claim 12 wherein repeatedly adding a method to the class file object for each method associated with a stub generated for a remote object includes:
determining a number of methods associated with the stub in a remote interface.

14. (Previously Presented): The method of claim 1 wherein adding code to the method includes:

repeatedly adding code for each method added to the class file.

15. (Previously Presented): The method of claim 1 wherein at least one of adding a method to the class file and adding code to the method includes:
generating a tree of statements.

16. (Previously Presented): The method of claim 15 wherein generating a tree of statements includes:

generating a tree representing at least one method, the at least one method comprising at least

one of: a code statement, an expression, a variable and a programming construct.

17. (Previously Presented): The method of claim 15 wherein generating a tree of statements includes:

generating a tree forming a known structure when the class file container is a known type.

18. (Previously Presented): The method of claim 17 wherein generating a tree forming a known structure when the class file container is a known type includes:

generating a tree forming a known structure when the class file container is of at least one of an adapter and a proxy type.

19. (Previously Presented): The method of claim 1 wherein generating byte code for the class file container object includes:

maintaining a state of a program being generated by each statement.

20. (Previously Presented): The method of claim 19 wherein maintaining a state of a program being generated by each statement includes:

maintaining at least one of a content of a stack, a content of a variable in use during program flow.

21. (Previously Presented): The method of claim 20 further comprising: generating an intermediate representation of program flow based upon the at least one of a content of a stack, a content of a variable in use during program flow.

22 - 24. (Cancelled)

25. (New): The method of claim 1, wherein the dynamically generated code is used for remote method invocation skeletons, remote method invocation stubs, wrappers for JDBC connections, and proxies used to enforce call-by-value semantics between EJBs.